**(Średnio)zaawansowane programowanie w C++ (ZPR2012L)**

**Dokumentacja funkcjonalna**

1. **Temat:**

Implementacja prostego symulatora ruchu miejskiego.

1. **Opis docelowego modelu:**

Podczas symulacji poruszają się obiekty różnego typu: samochody małe, duże oraz piesi. W różnych miejscach ulic rozmieszczone są "inteligentne" kamery - każda posiada swoje parametry: dokładność, kąt widzenia, kierunek obserwacji. Kamery są próbkowane co 1 sek. Jeżeli coś zauważy - generuje obserwacje i podaje współrzędne zaobserwowanych obiektów.

1. **Docelowa funkcjonalność:**
2. Pobieranie danych z plików konfiguracyjnych XML:
   * Mapa ulic w postaci węzłów grafu
   * Trasa samochodu wraz z typem w postaci kolejnych punktów
   * Trasa pieszych w postaci kolejnych punktów
   * Współrzędne kamer i ich parametry
3. Samochody różnią się od siebie parametrami (prędkość maksymalna, masa), rozpędzają się i zwalniają przed zakrętami (zależnie od promienia skrętu) - prosta fizyka
4. Piesi chodzą po liniach prostych, skręcają 'w miejscu'. Mogą poruszać się po całej mapie
5. Samochody skręcają po łukach – symulator automatycznie wyznacza płynne trasy przejazdu
6. Kamery mają swoje parametry - dokładność, kąt widzenia, kierunek obserwacji
7. Dane z kamer są logowane do pliku gdy podczas próbkowania w ich zasięgu widzenia znajdują się jakieś obiekty
8. Prosta wizualizacja graficzna symulacji w postaci obiektów geometrycznych
9. Możliwość zapętlenia symulacji - obiekty w kółko poruszają się po swoich trasach
10. Ze względu na ograniczenia czasowe:
    * Nie zakładamy kolizji obiektów - samochody i piesi przenikają się
    * Brak zaawansowanego GUI
11. **Funkcjonalność dodatkowa:**
12. GUI pozwalające ustalić trasy obiektów oraz położenie kamer za pomocą myszy
13. Weryfikacja poprawności wprowadzonych danych (pokrycie na mapie)
14. Sprawdzanie poprawności wprowadzonych ręcznie danych
15. **Wstępna organizacja projektu:**

Aplikacja składać się będzie z kilku głównych wątków. Wątek modelowanej symulacji, GUI, timer czuwający nad częstotliwością obliczeń modelu oraz próbkowaniem kamer, logger odpowiedzialny za generowanie do pliku danych z kamer.